CLAIMS

1. Method for dynamically optimize data throughput at the radio interfaces of a packet data cellular network having at disposal of said interfaces one or more type of modulations having different immunity from transmission errors when used for transmitting bursts of data, packed-up in blocks, between mobile stations (MS) and the serving base station (BTS), and vice versa, obtaining for each available modulation an upgrade and/or a downgrade tabulated threshold of the Block Error Rate, or BLER, delimiting a range in which that modulation outperforms the other available modulations in term of net data throughput; the BLER on the relevant temporary connection being continuously averaged and compared with the tabulated thresholds for selecting the proper modulation **characterized in that** includes the steps of:

- combining each available modulation with two or more coding schemes obtaining as many modulation-and-coding schemes, termed hereinafter MCSs, with different protection against transmission errors;
- obtaining for each MCS a first upgrade and/or a first downgrade tabulated BLER threshold (A) valid for low-diversity RF channels, delimiting a range in which that MCS outperforms the other available MCSs in term of net data throughput, considering as low-diversity a channel without frequency hopping and with low user mobility;
- obtaining for each MCS a second upgrade and/or a second downgrade tabulated BLER threshold (B) valid for high-diversity RF channels, delimiting a range in which that MCS outperforms the other available MCSs in term of net data throughput, considering as high-diversity a channel characterized by frequency hopping or high user mobility;
- selecting either the first (A) or the second (B) tabulated thresholds according to the diversity of the RF channel which sustains the temporary connection and use the selected thresholds for discriminating the right MCS.

2. Method for dynamically optimizing data throughput according to claim 1, characterized in that the step of discriminating the right MCS is performed by:

- updating at each new incoming block of data an averaged value of BLER evaluated in correspondence of the actual MCS;
- comparing said averaged BLER with the upgrade and/or downgrade thresholds of the actual MCS;
- replacing the actual MCS with the MCS immediately less error protected when the averaged BLER is lower than said upgrade threshold; or

– replacing the actual MCS with the MCS immediately more error protected when the averaged BLER is higher than said downgrade threshold.

3. Method for dynamically optimizing data throughput according to claim 1, characterized in that includes the step of obtaining for each MCS third upgrade and/or third downgrade tabulated BLER thresholds (C) valid for both low-diversity channels and incremental redundancy active, delimiting a range in which that MCS outperforms the other available MCSs in term of net data throughput.

4. Method for dynamically optimizing data throughput according to claim 1, characterized in that includes the step of obtaining for each MCS fourth upgrade and/or downgrade tabulated BLER thresholds (D) valid for both high-diversity channels and incremental redundancy active, delimiting a range in which that MCS outperforms the other available MCSs in term of net data throughput.

5. Method for dynamically optimizing data throughput according to claim 3, characterized in that the receiving entity carries out the steps of:

– temporarily storing errored data blocks in a memory buffer for joint decoding them with new transmissions of original blocks according to the incremental redundancy technique;
– continuously checking a condition of buffer full and other causes making retransmission with incremental redundancy inapplicable, for building a status variable, named hereinafter IR_status, which measures the averaged status of the incremental redundancy.

6. Method for dynamically optimizing data throughput according to the preceding claim when it depends on claim 3, characterized in that for each MCS a linear interpolation is performed run-time between first (A) and third (C) upgrade threshold and/or between first (A) and third (C) downgrade threshold, using the IR_status as interpolating factor for unbalancing the entity of the interpolation either towards third thresholds (C) when incremental redundancy prevails, or towards first thresholds (A) on the contrary case.

7. Method for dynamically optimizing data throughput according to claim 5, characterized in that for each MCS a linear interpolation is performed run-time between second (B) and fourth (D) upgrade threshold and/or between second (B) and fourth (D) downgrade threshold, using the IR_status as interpolating factor for unbalancing the entity of the interpolation either towards fourth thresholds (D) when incremental redundancy prevails, or towards second thresholds (B) on the contrary case.

- replacing the actual MCS with the MCS immediately more error protected when the averaged BLER is higher than said downgrade threshold.

3. Method for dynamically optimizing data throughput according to claim 1, characterized in that includes the step of obtaining for each MCS third upgrade and/or third downgrade tabulated BLER thresholds (C) valid for both low-diversity channels and incremental redundancy active, delimiting a range in which that MCS outperforms the other available MCSs in term of net data throughput.

4. Method for dynamically optimizing data throughput according to claim 1, characterized in that includes the step of obtaining for each MCS fourth upgrade and/or downgrade tabulated BLER thresholds (D) valid for both high-diversity channels and incremental redundancy active, delimiting a range in which that MCS outperforms the other available MCSs in term of net data throughput.

5. Method for dynamically optimizing data throughput according to claim 3, characterized in that the receiving entity carries out the steps of:

- temporarily storing errored data blocks in a memory buffer for joint decoding them with new transmissions of original blocks according to the incremental redundancy technique;
- continuously checking a condition of buffer full and other causes making retransmission with incremental redundancy inapplicable, for building a status variable, named hereinafter IR_status, which measures the averaged status of the incremental redundancy.

6. Method for dynamically optimizing data throughput according to the preceding claim when it depends on claim 3, characterized in that for each MCS a linear interpolation is performed run-time between first (A) and third (C) upgrade threshold and/or between first (A) and third (C) downgrade threshold, using the IR_status as interpolating factor for unbalancing the entity of the interpolation either towards third thresholds (C) when incremental redundancy prevails, or towards first thresholds (A) on the contrary case.

7. Method for dynamically optimizing data throughput according to claim 5, characterized in that for each MCS a linear interpolation is performed run-time between second (B) and fourth (D) upgrade threshold and/or between second (B) and fourth (D) downgrade threshold, using the IR_status as interpolating factor for unbalancing the entity of the interpolation either towards fourth thresholds (D) when incremental redundancy prevails, or towards second thresholds (B) on the contrary case.

8. Method for dynamically optimizing data throughput according to claim 5, characterized in that the averaged status of the incremental redundancy is obtained by weighting both the preceding and the actual values of a variable, named hereinafter IR_check, taking value 1 if incremental redundancy is properly working, or value 0 on the contrary, using a digital filter having a pulse response exponentially decreasing with discrete time n spanning a data block period.

9. Method for dynamically optimizing data throughput according to claim 1; preceding claims, characterized in that said averaged value of BLER is obtained by weighting both the preceding values of BLER and the actual decisions on errored blocks, using a digital filter having a pulse response exponentially decreasing with discrete time n spanning a block period.

10. Method for dynamically optimizing data throughput according to claim 9, characterized in that the pulse response of said digital filter of BLER is obtained by summing up two weight functions both accepting samples with the commanded MCS, a first one to weigh the preceding values of BLER and the second one to weigh the actual decisions on errored blocks.

11. Method for dynamically optimizing data throughput according to claim 10, characterized in that said first and second weight functions have balanced weights, so that an arbitrary increasing of the weight of the first function also involves an equal decreasing of the weight of the second function, and vice versa.

12. Method for dynamically optimizing data throughput according to claim 11, characterized in that the weight of said first and second weight functions are both equally varied in order to compensate the missing filtering effect of possible lacking blocks, in that making the outlined pulse response possible.

13. Method for dynamically optimizing data throughput according to claim 12, characterized in that the variation of said weights are carried out by making the said first and second weight functions further depending on a reliability function which tracks the age of the received blocks.

14. Method for dynamically optimizing data throughput according to claim 10, characterized in that said temporary connection is dedicated to transfer packet data from a selected mobile station to the base station, and said pulse response of BLER digital filter is obtained by means of the following function:

$$BLER_n = f_1(BLER_{n-1}) + f_2(s_n)$$

where:

- n is the iteration index spanning one block period;

- $s_n = 0$ if the block at instant n has been correctly received;
- $s_n = 1$ if the block at instant n has not been correctly received;
- $s_n = \frac{1}{K}\sum_{k=1}^{K} s_{n,k}$ if K blocks are received for the considered connection;
- $f_1(BLER_{n-1})$ is said first weight function, taking values inside the interval 0 - 1;
- $f_2(s_n)$ is said second weight function of the variable s_n relative to the decision on the errored blocks, taking values inside the interval 0 - 1;

15. Method for dynamically optimizing data throughput according to the preceding claim, characterized in that said first and second weight functions assume the following expressions:

$$f_1(BLER_{n-1}) = (1-\beta\cdot\frac{x_n}{R_n})\cdot BLER_{n-1}$$

$$f_2(s_n) = \beta\cdot\frac{x_n}{R_n}\cdot s_n$$

where:

- x_n is equal to 1 if "at least" one RLC block for the considered connection with the commanded MCS is received at time instant n, otherwise is set to 0;
- $\beta = 1/T_{AVG}$ is a forgetting factor and T_{AVG} being the filtering period in multiples of a radio block;
- $R_n = (1-\beta)\cdot R_{n-1} + \beta\cdot x_n;\quad R_{-1} = 0$ is said reliability function.

16. Method for dynamically optimizing data throughput according to claim 8, characterized in that said temporary connection is dedicated to transfer packet data from a selected mobile station to the base station, and said pulse response of the digital filter of the IR_status is obtained by means of the following function:

$$IR\_status_n = f_1(IR\_status_{n-1}) + f_2(IR\_check_n)$$

were:

- n is the iteration index spanning one block period;
- f_1 and f_2 are weight functions following the same laws as used in the BLER calculation.

17. Method for dynamically optimizing data throughput according to claim 16, characterized in that said first and second weight functions assume the following expressions:

$$f_1(IR\_status_{n-1}) = (1-\beta\cdot\frac{x_n}{R_n})\cdot IR\_status_{n-1}$$

$$f_2(IR\_check_n) = \beta\cdot\frac{x_n}{R_n}\cdot IR\_check_n$$

where: R_n takes a formal expression as that used in the BLER calculation, while x_n and β are the same.

18. Method for dynamically optimizing data throughput according to claim 6, characterized in that said linear interpolations take the following expressions:

$UP\_th_n = (1 - IR\_status_n) \times BLER_{MCSx\rightarrow MCSy} + IR\_status_n \times BLER_{MCSx\_wIR\rightarrow MCSy\_wIR}$

$DN\_th_n = (1 - IR\_status_n) \times BLER_{MCSx\rightarrow MCSz} + IR\_status_n \times BLER_{MCSx\_wIR\rightarrow MCSz\_wIR}$

where:

- UP_th_n and DN_th_n are the upgrade and downgrade thresholds, respectively, at the n-th block period;
- $BLER_{MCSx\rightarrow MCSy}$ is an upgrade first (A) or second (B) tabulated threshold;
- $BLER_{MCSx\_wIR\rightarrow MCSy\_wIR}$ is an upgrade third (C) or fourth (D) tabulated threshold;
- $BLER_{MCSx\rightarrow MCSz}$ is a downgrade first (A) or second (B) tabulated threshold;
- $BLER_{MCSx\_wIR\rightarrow MCSz\_wIR}$ is a downgrade third (C) or fourth (D) tabulated threshold.

19. Method for dynamically optimizing data throughput according to claim 10, characterized in that said temporary connection is dedicated to transfer packet data from the base station to a selected mobile station, and said pulse response of BLER digital filter is obtained by means of the following function:

$$BLER_k = f_1(BLER_{k-1}) + f_2(s_k)$$

where:

- k is the reporting instant lasting m blocks;
- $s_k = \frac{Nack\_blocks}{Sent\_blocks}$

  Nack_blocks: number of badly received blocks among those sent with the present MCS;

  Sent_blocks: number of blocks sent with the present MCS in the previous polling period:
- $f_1(BLER_{k-1})$ is said first weight function, taking values inside the interval 0 - 1;
- $f_2(s_k)$ is said second weight function of the variable s_k relative to the decision on the errored blocks, taking values inside the interval 0 - 1.

20. Method for dynamically optimizing data throughput according to claim 19, characterized in that said first and second weight functions assume the following expressions:

$$f_1(BLER_{k-1}) = (1-\frac{\beta}{R_k})\cdot BLER_{k-1}$$

$$f_2(s_k) = \frac{\beta}{R_k}\cdot s_k$$

where:

- $\beta = 1/T_{AVG}$ is a forgetting factor and T_{AVG} being the filtering period in multiples of a radio block;
- $R_k = (1-\beta)^m \cdot R_{k-1} + \beta;\ R_{-1} = 0$ is said reliability function.

21. Method for dynamically optimizing data throughput according to claim 19, characterized in that said temporary connection is dedicated to transfer packet data from the base station to a selected mobile station, and said pulse response of IR_status digital filter is obtained by means of the following function:

$$IR\_status_k = f_1(IR\_status_{k-1}) + f_2(IR\_check_k)$$

were:

- k is the reporting instant lasting m blocks;
- f_1 and f_2 are weight functions following the same laws as used in the BLER calculation.

22. Method for dynamically optimizing data throughput according to claim 21, characterized in that said first and second weight functions assume the following expressions:

$$f_1(IR\_status_{k\text{-}1}) = (1-\frac{\beta}{R_k})\cdot IR\_check_{k-1}$$

$$f_2(IR\_check_k) = \frac{\beta}{R_k}\cdot IR\_check_k$$

where: R_k takes a formal expression as that used in the BLER calculation, and β is the same.

23. Method for dynamically optimizing data throughput according to claim 1, characterized in that a modified power control works in parallel with the MCS switching link adaptation and the modified power control includes the following steps:

- off-line calculation of the expression:

$$T_{PxTS} = T_P / N_{TS},$$

where: T_{PxTS} is the Peak Throughput per timeslot; T_P is the Peak Throughput derived from the Quality of Service Class of the connection, and N_{TS} is the minimum between the number of allocable timeslots and the number of timeslots that can be handled by the mobile station due to its multislot class;

- off-line mapping of the calculated T_{PxTS} on a simulated curve depicting the maximum achievable net throughput in function of the values of Carrier versus Interference C/I, and obtaining from the curve a target C/I_{target} value;
- exploiting the C/I_{target} for all the duration of the ongoing connection as a goal to be maintained by the network (BSC, BTS) exploiting the Power and Interference measures at the receiver side.